CITS4407 Open Source Tools and Scripting
Semester 1, 2021
Week 5 workshop – Text and regular expressions

Before starting this workshop, make sure you've reviewed the recommended reading for weeks 1–4, and completed the lab sheets for weeks 2–4.

0. Revision

In your reading from previous weeks, you should have read about the `sort`, `uniq`, `wc`, `head` and `tail` commands. (If not – refer back to chapter 6, "Redirection", of *Shotts*.)

Answer the following questions. It is suggested you try these problems on your own and discuss with a partner – then compare your answers with the rest of the class.

Counting

```
$ echo a b c
```

What is a pipeline we can add to the end of this command (i.e., the pipe character, "`|`", plus further commands) which will display how many words the `echo` command has printed? How does it work?

Duplicates

Executing the following

```
ls /usr/bin /bin
```

will list the contents of the `/usr/bin` and `/bin` directories, which contain executable programs for external commands.

Traditionally on a Unix-like system, `/bin` contains just a subset of the typical Unix commands – it contains only those commands needed by the system for booting and potentially repairing the system. Commands in `/usr/bin/`, on the other hand, are not needed for booting or repair, and thus could be kept on a separate disk and loaded later.

Some programs appear only in `/bin`, some only in `/usr/bin`, and some appear in both directories.

What is a pipeline we can add to the command shown, which will print the programs found in both directories? How does it work?

Challenge exercise: How can we print the files that are in `/bin`, but not `/usr/bin`?

1. Shell expansions

It is suggested you try these problems on your own and discuss with a partner – then compare your answers with the rest of the class.

Hidden files

Try running the following in your Bash terminal:

```
$ touch .hidden-file
$ ls -a
$ echo *
```

What do you expect to see, and why? Can you suggest a filename expansion that you can put after "`echo`", that will print `.hidden-file` (and possibly other dot-files in the working directory)?

Arithmetic expansion

Look at the following commands:

```
$ touch file1
$ myvar="INSERT SOMETHING HERE"
$ echo file$((4 - myvar))
```

If you run these, you will just get an error. But assigning the correct value to `$myvar` will result in `file1` being printed by the `echo` command. What value do you need to insert, and why does this work?

Command substitution

Run the following commands:

```
$ echo "some content" > file1
$ echo "$(cat file1)"
```

From previous weeks, you should know what the first line is doing – it is writing the output of the `echo` command to `file1`.

The next line is performing a type of expansion called *command substitution*. The dollar sign tells bash that the string between the parentheses should be interpreted as commands to run, and the dollar and parentheses should be *replaced* by the output of those commands.

```
$ echo "some content" > file1
```

```
$ echo "$(cat file1)"
```

Look at the following command:

```
$ echo "The location of the cp command is: XXX"
```

Suppose we wish to replace the `XXX` with the location of the `cp` command – what should we put there, and why?

Quoting

Which of the following will print a single asterisk to the screen? Why?

a. `echo *`
b. `echo "*"`
c. `echo '*'`

2. Scripts

If you haven't tried the exercises in *Shotts* chapter 20 ("Text processing"), attempt those first before doing this section.

It is suggested you try these problems on your own and then discuss with a partner – then compare your answers with the rest of the class.

Running scripts

Try running the following commands:

```
$ echo "echo hello" > my-script.sh
$ ./my-script.sh
```

You should see an error message. What is the problem here, and how do we fix it?

Manipulating text

Clone the Git repository at `<https://github.com/cits4407/workshop04>` onto your computer. The file contained in it stores data on enrolments at Australian universities, and is in *tab-separated* format – each line contains several "fields", and the fields are separated by tab characters. Files like this are often given the extension `.tsv`.

(**Challenge exercise:** Can you find a way of downloading the `.tsv` file without using Git? Investigate the `wget` and `curl` commands, and take a look at the Github repository in your browser to see what you can access via the browser.)

Write a script that uses the `sort` command to print the line of the file which contains a record of the university with the highest total enrolment. (Hint: you might want to use the `head` command, as well – check the documentation for it.)

Then extend your script to also print the lines for the universities with the highest and lowest local enrolment.

Finally, add pipelines in your script, using the `cut` command to print just the *names* of the universities, not the whole line.